

**RECEIVED**

AUG 13 2003

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE** Technology Center 2100

Applicant: Seiki Aguro  
Serial No.: 09/281,042  
Filed: 03/30/1999  
For: COMPUTER SYSTEM

Docket No.: TIJ-26495  
Art Unit: 2123  
Examiner: Jones, H.  
Confirm. No.: 6678

**DECLARATION TO OVERCOME REJECTION UNDER 35 U.S.C. 112,  
FIRST PARAGRAPH (37 CFR § 1.132)**

Commissioner of Patents  
P. O. Box 1450  
Alexandria, VA 22313-1450

**MAILING CERTIFICATE UNDER 37 C.F.R.  
§ 1.8(A)**

I hereby certify that the above correspondence is being deposited with the U.S. Postal Service as First Class Mail in an envelope addressed to: Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450.

*William B. Kempler*  
William B. Kempler

*8/8/03*  
DATE

Dear Sir:

1. I, Yasuhiro Ikeyoshi, an electrical engineer having received a 4 year bachelor's degree from the National Defense Academy in 1989. I have 4 years of experience in embedded firmware development for storage devices and 3 years of experience in LSI device evaluation for storage devices, as well as other experience in the electrical engineering field. I am currently employed by Texas Instruments Japan, Ltd. as a DVD-DFE and DCES application engineer.

2. On 07/04/2003 I was asked to read a U.S. patent application, serial number 09/281,042, entitled "Computer System" and prepare a computer program to obtain the

TU-26495

password for the device of the patent application, including obtaining parts of the password presented at different ports of the processor at different times, and comparing it to a stored password for allowing access to the JTAG information. My only source of information on what is required, is from the patent application itself.

3. I was able to read the patent application and prepare the computer program attached hereto, written in the C language, in about 1 hour. Although this program, as with any computer program, may be subject to being debugged, I believe that it can readily be made to perform this task using only routine measures.

4. This declaration is submitted prior to final rejection.

5. As a person signing below:

I hereby declare that all statements made herein on my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Respectfully submitted,



Yasuhiro Ikeyoshi

Date: 08/01/2003

```
/*-----*/
#define kCMD_CODE_SECURITY_ON    0xCBD1
#define kCMD_CODE_SECURITY_OFF  0xB0CD
#define kCMD_CODE_SECURITY_KEY  0x57BF

#define kPW0_Key                 0x3586
#define kPW1_Key                 0xEE42

#define adrPW0                   0x0060
#define adrPW1                   0x0061

#define adrCMD_CODE              0x0020
#define adrCMD_DESCRIPTOR_0      0x0022
#define adrCMD_DESCRIPTOR_1      0x0023
#define adrCMD_DESCRIPTOR_2      0x0024
#define adrCMD_DESCRIPTOR_3      0x0025

#pragma interrupt cmd_int

int  cmd_code = 0;

/* Interrupt Handler : Receive Command from the host */
void cmd_int(void)
{
    cmd_code = *(volatile int *)(adrCMD_CODE);
}

/* Disable protection by the unique key */
void disable_protection(void)
{
    *(volatile int *)(adrPW0) = kPW0_Key;
    *(volatile int *)(adrPW1) = kPW1_Key;
}

/* Enable protection */
void enable_protection(void)
{
    *(volatile int *)(adrPW0) = \
    *(volatile int *)(adrPW1) = 0;
}

/* Change protection state by direct key exchange */
void direct_protection(void)
{
    int local_key_0, local_key_1;

    /* CMD_DESC_0 XOR CMD_DESC_3 */
}
```

```
local_key_0 = \
    *(volatile int *) (adrCMD_DESCRIPTOR_0) ^ \
    *(volatile int *) (adrCMD_DESCRIPTOR_3);

/* CMD_DESC_1 XOR CMD_DESC_2 */
local_key_1 = \
    *(volatile int *) (adrCMD_DESCRIPTOR_1) ^ \
    *(volatile int *) (adrCMD_DESCRIPTOR_2);

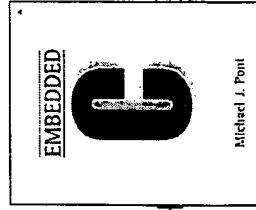
*(volatile int *) (adrPW0) = local_key_0;
*(volatile int *) (adrPW1) = local_key_1;
}

/* Main program */
int main(void)
{
    while(1){
        switch( cmd_code ){
            case kCMD_CODE_SECURITY_OFF :
                disable_protection();
                cmd_code = 0;
                break;
            case kCMD_CODE_SECURITY_ON :
                enable_protection();
                cmd_code = 0;
                break;
            case kCMD_CODE_SECURITY_KEY :
                direct_protection();
                cmd_code = 0;
                break;
            default ;
        }
        ;
        /* other code */
        ;
    }
}

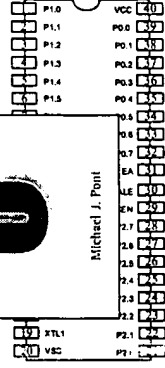
/* $end -----*/
```

# Programming Embedded Systems I

A 10-week course, using C



Michael J. Pont  
University of Leicester

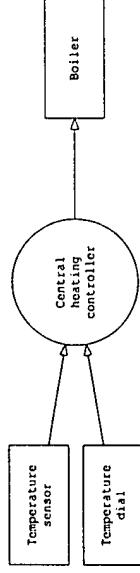


[v1.2]

Copyright © Michael J. Pont, 2002-2003

This document may be freely distributed and copied, provided that copyright notice at the foot of each OHP page is clearly visible in all copies.

## Example: Central-heating controller



```
void main(void)
{
    /* Init the system */
    C_HEAT_Init();
    while(1) /* 'for ever' (Super Loop) */
    {
        /* Find out what temperature the user requires
         (via the user interface) */
        C_HEAT_Get_Required_Temperature();

        /* Find out what the current room temperature is
         (via temperature sensor) */
        C_HEAT_Get_Actual_Temperature();

        /* Adjust the gas burner, as required */
        C_HEAT_Control_Boiler();
    }
}
```

## Reading from (and writing to) port pins

### Problem

How do you write software to read from and /or write to the ports on an (8051) microcontroller?

### Background

The Standard 8051s have four 8-bit ports.

All of the ports are bidirectional: that is, they may be used for both input and output.